

Citation for published version:

Cliff, P, Kelly, B & Powell, A 2001, 'RDN-Include: Re-branding Remote Resources', WWW10, Hong Kong, 1/05/01 - 5/05/01.

Publication date:
2001

[Link to publication](#)

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

RDN-Include: Re-branding Remote Resources

Peter Cliff
UKOLN
University of Bath
Bath, UK
+44 1225 323218
p.d.cliff@ukoln.ac.uk

Brian Kelly
UKOLN
University of Bath
Bath, UK
+44 1225 323943
b.kelly@ukoln.ac.uk

Andy Powell
UKOLN
University of Bath
Bath, UK
+44 1225 323933
a.powell@ukoln.ac.uk

ABSTRACT

The Resource Discovery Network (RDN) has developed tools that allow educational institutions and other organizations to access and display the output from its Web services with their own branding and look-and-feel. This paper describes the need for such tools and the architectural approach to their development. A potential application of this approach to the incorporation of news feeds is also given.

Keywords

Resource discovery, subject gateways, content sharing, re-use, news feeds, RSS, XML.

1. BACKGROUND

The Resource Discovery Network (RDN) [1] is a national service funded by JISC to provide access to high quality Internet resources for the UK Higher and Further Education communities. The RDN is a cooperative network of subject gateways, including SOSIG [2] (social science, business and law), EEVL [3] (engineering, mathematics and computing), BIOME [4] (health, medicine and life sciences), HUMBUL [5] (humanities) and PSIgate [6] (physical sciences).

Individual gateways provide search and browse access to databases containing descriptions of high quality Internet resources in their subject area as well as other community-specific services. Although each gateway presents its own Web-based user-interface, they all work within a shared architectural framework that supports interoperability between the services, including the ability to cross-search all the RDN gateways. This results in increased functionality for the end-user and the institutions that make use of RDN services.

2. THE INSTITUTION'S VIEW

The subject gateways have proved popular with the end-user community, and with those institutions that understand the resource commitments required in order to maintain links to high quality resources [7]. However, as Web services play a greater role in their marketing, those institutions are beginning to show increasing concern over the branding of their Web services and a reluctance to provide links to external services which may be perceived (albeit incorrectly) to be in competition with the services they provide. Furthermore, for end-users there is some benefit in being able to access remote services, such as the RDN, in such a way that they retain the institutional look-and-feel with which they are accustomed.

3. THE RDN-INCLUDE SOLUTION

RDN-Include [8] has been developed to enable institutions to provide access to the RDN cross-search and top-level browse interface, whilst maintaining an institutional look-and-feel and URL. RDN-Include allows the institution to embed an RDN search form and top-level

browse hierarchy into appropriate points in their Web site. Search results are also returned with an institutional look-and-feel.

To make use of RDN-Include the institution must install the RDN-Include CGI script on one of their Web servers. All end-user interaction is handled by this script, thus retaining an institutional URL. If the script is not passed any search parameters, it reads a template file held locally. The template file is essentially a normal HTML or XHTML Web page, formatted with a local look-and-feel. RDN content (the RDN search box and browse hierarchy) is added to this template dynamically at run time by including in it one or more XML 'rdni:tag' elements of the form:

```
<rdni:tag val="RDN*" />
```

where * represents a unique service identifier. These tags are defined in a tag definition file at the RDN. Based on these tags, the RDN-Include CGI script collects appropriate content from the RDN Web server using HTTP. In this way the RDN can update and/or create tags as new services are developed. Any new tags immediately become available to RDN-Include hosts, so the RDN can add new services without the need to update the RDN-Include script installed on each institutional machine.

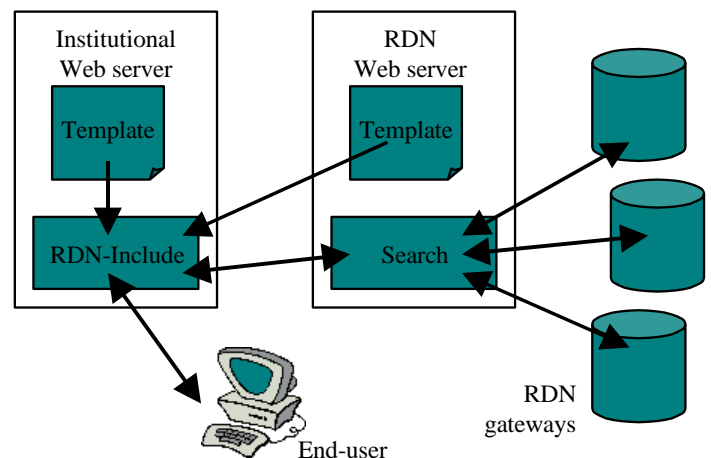


Figure 1: RDN-Include

If search parameters are passed to the RDN-Include script, it passes these on to a CGI search script running on the RDN Web server. This script cross-searches all the available RDN gateways, using the search protocols defined in the RDN Interoperability and Standards Framework [9], and returns results as 'plain' HTML that can be embedded within the institutional template.

RDN-Include has been implemented as both a Perl CGI script and using ASP.

This approach is:

- *Simple*: installing RDN-Include only requires access to a Web server that supports CGI Perl or ASP,
- *Robust*: failure of the RDN Web server (or network problems) does not prevent the institutional content of the page being displayed,
- *Flexible*: control over the content of the page is shared between the institution and the RDN,
- *Scalable*: the RDN can offer new functionality by developing new RDN-Include 'rdni:tag' elements.

4. THE RDNi-LITE SOLUTION

One problem with the approach provided by RDN-Include is the need for the support of the Web server administrator, since it requires the installation of a CGI Perl or ASP script. System administrators have legitimate concerns over the security of third-party scripts. In addition there may be local political barriers to the deployment of such scripts.

An alternative approach is to develop an RDN-Include solution that can be deployed directly by HTML authors. RDNi-Lite provides this solution through the use of JavaScript.

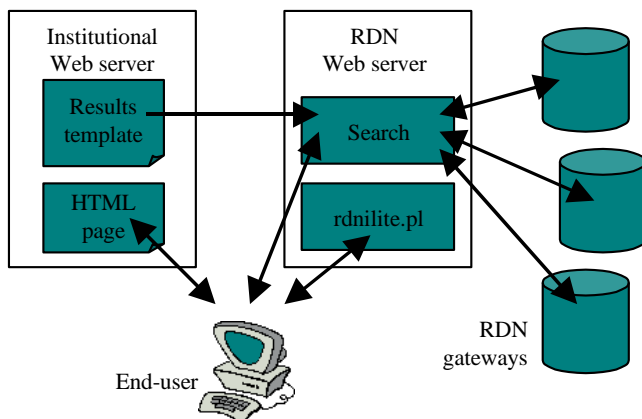


Figure 2: RDNi-Lite

To make use of RDNi-Lite, an HTML author simply has to embed a Javascript `<script>` element into a normal HTML page. The 'src' attribute of the script element points to a CGI Perl script running on the RDN Web server. The CGI script generates a set of `document.write()` JavaScript statements that, when executed by the Web browser, embed the appropriate RDN content into the Web page. The CGI script takes several optional arguments that provide 'rdni:tag' parameters. The search form generated by the script calls a separate search CGI script hosted by the RDN that will, optionally, gather a template file from the local Web server in order to format results with a particular look-and-feel.

This approach means that there is no need for the owner of the Web page to have access privileges to install a CGI Perl or ASP script. However, it does require that Javascript be enabled in the Web browser.

5. APPLICATION TO NEWS FEEDS

In a related set of activities, we have developed a Web service called RSS-xpress [10], which provides a suite of CGI tools for creating, updating, parsing and viewing RDF Site Summary (RSS) [11] files. RSS is an XML application for sharing metadata about 'news' content in the form of RSS channels. In many cases, adding support for remote RSS channels to your Web site involves installing and configuring a CGI RSS viewer software that gathers one or more remote RSS channels and displays them as HTML (typically embedded within other local content). We are currently working on a lightweight mechanism for embedding RSS channels in HTML pages using the same techniques adopted for RDNi-Lite. By using a Javascript `<script>` element that points to a remote CGI-based RSS parser, HTML page authors will be able to embed 'news' content into their pages dynamically without needing to install Perl scripts, PHP, ASP or other local scripting technologies.

6. CONCLUSIONS

Our work on RDN-Include has convinced us of the benefits of offering RDN content in a form that allows others to embed it within local services. However, even the minimal technical requirements of RDN-Include present a significant hurdle for some service providers. It is hoped that the development of lightweight solutions that can be used directly by HTML authors, such as RDNi-Lite, will overcome some of these problems and encourage the use of shared content.

7. REFERENCES

- [1] Resource Discovery Network, <http://www.rdn.ac.uk/>
- [2] SOSIG, <http://www.sosig.ac.uk/>
- [3] EEVL, <http://www.eevl.ac.uk/>
- [4] BIOME, <http://biome.ac.uk/>
- [5] HUMBUL, <http://www.humbul.ac.uk/>
- [6] PSIGate, <http://www.psigate.ac.uk/>
- [7] DESIRE Information Gateways Handbook <http://www.desire.org/handbook/>
- [8] RDN-Include: Why send users to the RDN when the RDN can come to you? <http://www.rdn.ac.uk/rdn-i/>
- [9] RDN Interoperability and Standards Framework <http://www.rdn.ac.uk/publications/interop-standards>
- [10] RSS-xpress <http://rssxpress.ukoln.ac.uk/>
- [11] RDF Site Summary (RSS) 1.0 <http://purl.org/rss/1.0/>